

9 – Databáze – jedná se o soubor obsahující buďto jiné soubory či data, tyto data mohou být jakéhokoliv typu ať text, číslo či obrázek. Nejobecnější databáze je tabulka skládající se ze sloupců a řádků. Sloupce nazýváme Atributy a řádky jsou Záznamy.

Databáze je také zabalena do SŘBD (systém řídicí bázi dat – více zmiňován po modelech databází) a ten je dále obalen v aplikaci která slouží jako uživatelské prostředí pro ulehčení práce s databází.

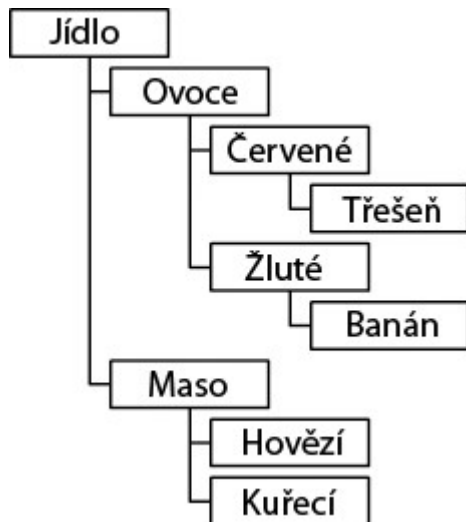
Data bázní pojmy:

- Atribut = sloupec tabulky
- Záznam = řádek tabulky
- Pole = buňka tabulky (místo kde se protíná sloupec a řádek)
- Primární klíč = jednoznačný(unikátní) identifikátor záznamů v tabulce
- Cizí klíč = klíč tabulky, který odkazuje na primární klíč jiné tabulky(vezmeš primární klíč jedné tabulky a uděláš druhou tabulku se stejným klíčem a ten zkopírovaný klíč je ten cizí)
- Relace = propojení mezi tabulkami, jenž vede k předávání informací mezi jednotlivými tabulkami
- Datový typ = určuje jaký druh data je v poli INT, TEXT atd (více definováno na další stránce)

Typy Relací: (během relace musí mít oba klíče stejný datový typ)

1. Jedna ku jedné (1-1) tabulky mají propojené primární klíče a obě referují pouze jednu hodnotu. Příklad: 1 dřevorubec kácí 1 strom a 1 strom je kácen 1 dřevorubcem. Tento typ relace se používá na zúhlednění jedné z tabulek či vzniká jako chyba špatné relace.
2. Jedna ku mnoho (1-N) tabulky mají opět propojené primární klíče, avšak jedním směrem tabulka referuje několik záznamů. Příklad: 1 dřevorubec rubí 1 strom, ale 1 strom může rubat vícero dřevorubců.
3. Mnoho ku mnoho (M-N) tabulky (přičemž nečekaně propojené primární klíči), ale předešlá několikanásobná reference je oběma směry. Příklad: 1 dřevorubec rubí několik stromů a 1 strom rubí několik dřevorubců.

Modely databází – Hierarchická – data jsou uspořádávány podle typu a později jemněji filtrovány



Síťová – hierarchický model s přidáním mnohonásobnými vztahy (sety), záznamy mohou mít stejné hodnoty a naopak (voe je to jak klíče nebo takata relační sračka že místo záznamu tam mrdneš proměnou)

Relační – nejmladší model, data jsou ukládány do tabulek, ty se skládají z řádků a sloupců (literárně to co děláme prostě popiš mySql)

Objektová – data jsou ukládány do objektů, ty mají svou identity, nepoužívá se primární klíč, a korespondují s věcmi z reálného světa. Objekty se dají propojovat a dědit. (představ si OOP jako databázi)

Objektově relační – relační model s přidáním objekty

- Systém Řízení Baze Dat (SŘBD) jsou souhrny procedur a datových struktur, ty zajišťují nezávislost, bezpečnost a modifikaci údajů. Pracují s databází.
- Databázové systémy – databáze jsou ukládány pomocí aplikací a také je možno je otevřít pouze pomocí těchto aplikací. Př Excell, Visual Basic, phpMyAdmin
- Dotazy mySQL – SELECT * FROM (databáze) select vybírá a vypíše požadované záznamy z databáze, hvězdička znamená že chceme všechny záznamy, pokud bychom chtěli specifické informace ze záznamů zapišeme to způsobem `SUM(Trzba)` `AS Soucet` slovíčko As nám uchová výpisy z příkazu select jako pole pod proměnou Součet, From udává odkud tyto záznamy bereme.
 - Dotaz můžeme doplnit i slovem Where – to funguje jako podmínka a vypíše nám tedy pouze záznamy splňující tuto podmínku
 - Taktéž se dá doplnit o ORDER to seřadí podle požadavku
 - UPDATE kde se dají SET a WHERE, které změni název sloupců či informací v nich
 - CREATE TABLE vytvoří tabulku s libovolným názvem a do závorek zatím vkládáme názvy a datové typy sloupců.
 - INSERT INTO zvolí sloupec a do závorek se vloží hodnoty, které vkládá do sloupce
 - DELETE FROM zapišeme název tabulky a pomocí WHERE zvolíme jaké data z databáze smažeme
 - DROP TABLE/DATABASE/ATTRIBUTE používá se ke smazání buďto celé tabulky, databáze či atributu (sloupce)

- Datová pole/typy: INTEGER – číslo, BOOLEAN – true nebo false, DECIMAL – desíma
- Čárka
 - TEXT – text
 - VARCHAR – znakový řetězec do 255 bajtů
 - DATE – datum, TIME – čas
 - SPATIAL – geometrické obrazce
- Optimalizace databází
 - Zkontrolujeme, zda je použitý datový model nejlepší pro danou databázi
 - Použijeme nejmenší možný datový typ, ten určíme podle největší hodnoty
 - Indexovat databázi pro zrychlení hledání (indexy selectnou data a ty si předpřipraví k vypsání)
 - Partitioning – rozdělit tabulku na sérii menších tabulek, při vyhledávání se zaměříme na specifickou část tabulky spíše než na celou databázi
 - Škálovat fyzický server (kup lepší RAMKY lol)
- Normálové formy:
 - Nultá normální forma – relace je v nulté podobě automaticky ve chvíli kdy máme vytvořenou databázi jelikož jedinou podmínkou je že aspoň jeden atribut musím mít více než jednu hodnotu
 - První normální forma – relace je v první formě když platí:
 - Atributy se nedají dále dělit
 - Jedinečné řádky
 - Přístup k řádkům relace podle obsahu atributů
 - Druhá normální forma:
 - Splňuje první
 - Neklíčové atributy nejsou parciálně závislé na žádném z klíčů z relace např:

Druhá normální forma

Například relace {IdStudenta, IdPredmetu, JmenoStudenta, Semestr} není ve 2NF, neboť:

- Klíčem je {IdStudenta, IdPredmetu}
- Neklíčový atribut JmenoStudenta je funčně závislý na {IdStudenta}, je tedy parciálně závislý na klíči.
- Neklíčový atribut Semestr je funčně závislý na {IdPredmetu}, je tedy parciálně závislý na klíči.

Řešení:

Relaci rozdělíme do tří tabulek

- {IdStudenta, IdPredmetu}
- {IdStudenta, JmenoStudenta}
- {IdPredmetu, Semestr}

- Třetí Normální forma:

- Splňuje 2 formu - Neklíčové atributy nejsou transitivně závislé na žádném z klíčů z relace (atribut nesmí být závislý na atributu, který je závislý na klíči)